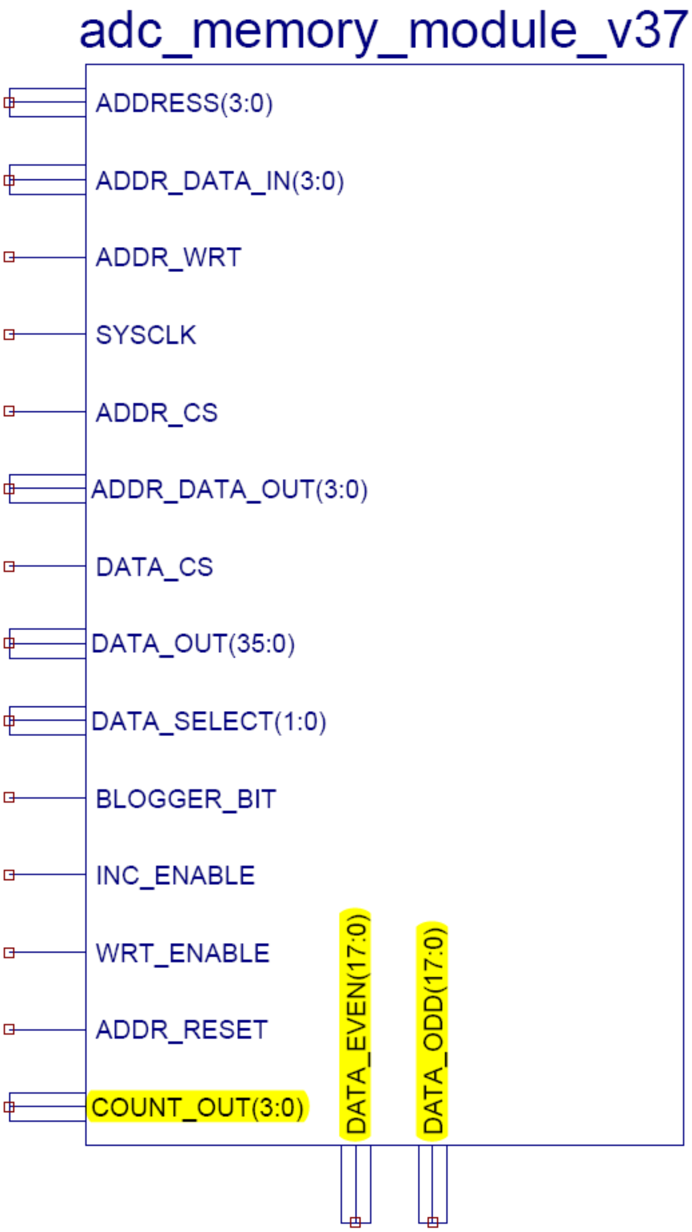
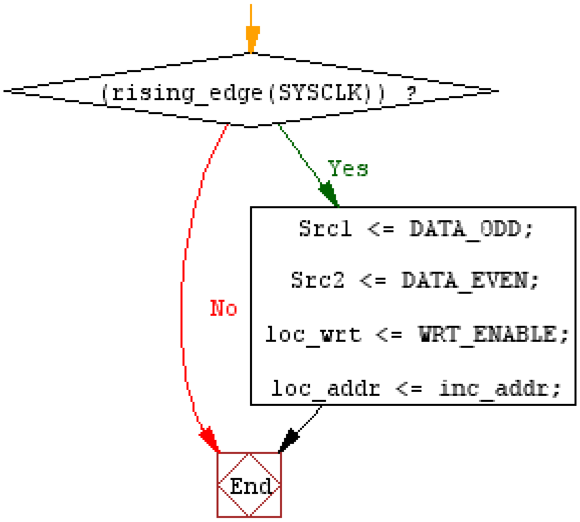
```
--- DESCRIPTION  :
--- THIS MODULE IMPLEMENTS A DATA STORE FOR ADC DATA. THE DATA IS STORED ON A
--- 16 DEEP x 24 BIT WIDE DUAL PORT RAM. ADC DATA IS WRITTEN SIMULTANIOUSLY
--- THROUGH BOTH PORTS TO TWO CONSECUTIVE LOCATIONS. THE EVEN RAM LOCATIONS
--- CONTAIN THE EVEN CHANNEL DATA AND THE ODD .... INCOMING DATA IS SUMMED
--- TO EXISTING DATA IN RAM (FROM A PREVIOUS CONVERSION CYCLE) WHEN SUM_ENABLE
--- IS TRUE. ONCE ACQUISITION IS COMPLETE, DATA MAY BE READ DIRECTLY FROM THE
--- RAM STORE BY THE PAN IN A NORMAL READ CYCLE. DATA IS TRUNCATED ACCORDING TO
--- THE NUMBER OF SUMMED VALUES AND OUTPUT AS 16 BIT DATA. IN ADDITION, THE
--- PIPELINE WRITE FSM CAN READ THE DATA OUT AND TRANSMIT THESE DATA TO THE PAN
--- AUTOMATICALLY. THIS TYPE OF READ IS PERFORMED THROUGH INDIRECTION TO ENABLE
--- CHANNEL SELECTION TO BE IMPLEMENTED. THE INDIRECTION IS VIA A SMALL RAM STORE
--- THAT IS FILLED WITH CHANNEL POINTERS BEFORE ACQUISTION STARTS. THIS RAM ACTS
--- AS A LOOKUP TABLE AND SORTS THE CHANNELS ACCORDING TO THE ORDER IN THE ADDR RAM.
-------------------------------------------------------------------------
--- IMPLEMENTATION NOTES :
--- BLOGGER BIT IS USED WHEN THE 'ChanCount' HAS A VALUE OF 1. THIS ALLOWS SINGLE
--- CHANNEL ACQUISITION BY MULTIPLEXING THE SHADOW RAMS (RamStore3 & 4) BETWEEN
--- TWO ACQUISITION CYCLES. THE PIPELINE WRITE FUNCTION THEN TRIGGERS EVERY TWO
--- ACQUISITION CYCLES TO SEND THE DATA TO THE PAN.
```

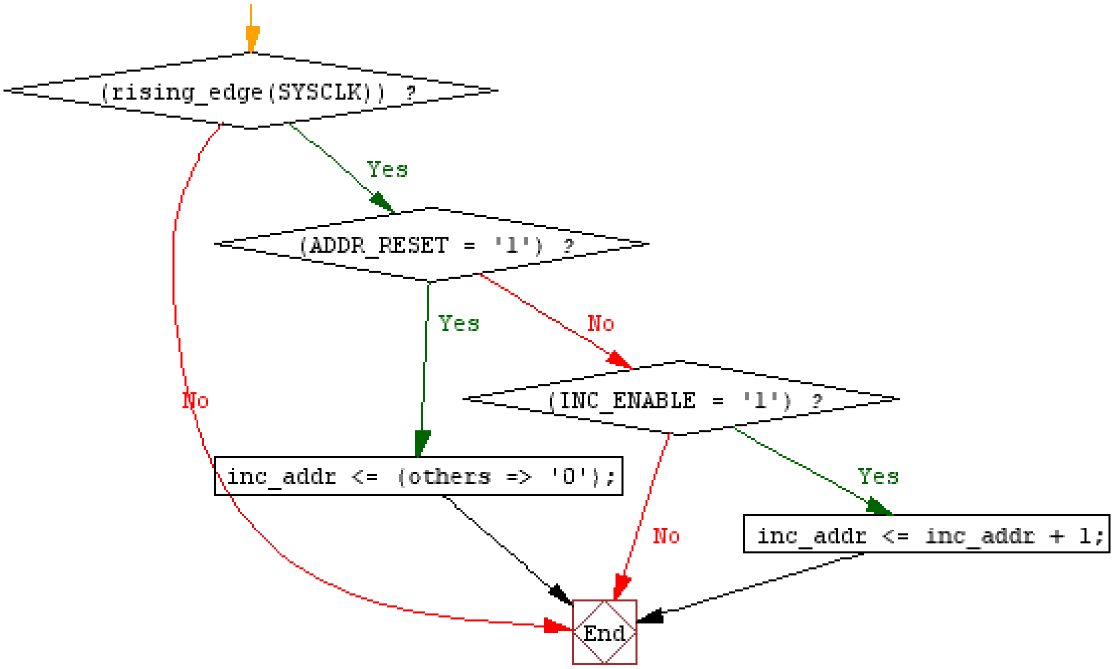


```
SYSCLK        : in  std_logic;
ADDRESS       : in  std_logic_vector(3 downto 0);
DATA_CS       : in  std_logic;
ADDR_CS       : in  std_logic;
ADDR_WRT      : in  std_logic;
ADDR_DATA_IN  : in  std_logic_vector(3 downto 0);
ADDR_DATA_OUT: out std_logic_vector(3 downto 0);
ADDR_RESET    : in  std_logic;
WRT_ENABLE    : in  std_logic;
INC_ENABLE    : in  std_logic;
BLOGGER_BIT   : in  std_logic;
DATA_SELECT   : in  std_logic_vector(1 downto 0);
DATA_EVEN     : in  std_logic_vector(17 downto 0);
DATA_ODD      : in  std_logic_vector(17 downto 0);
DATA_OUT      : out std_logic_vector(35 downto 0);
COUNT_OUT     : out std_logic_vector(3 downto 0)
```

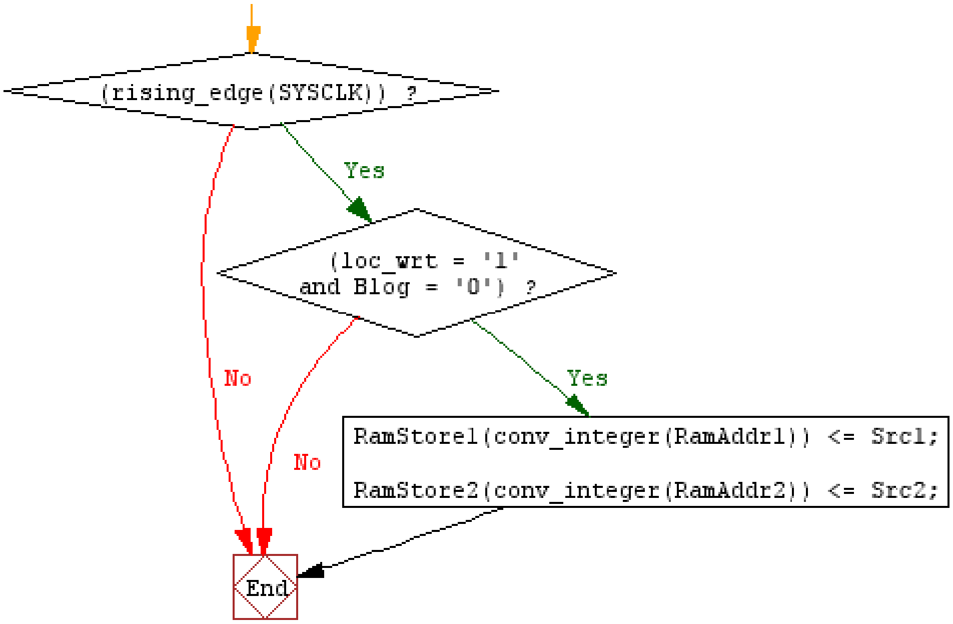
LocSumWrtPipeDly : process(SYSCLK)



LocRamAddr : process (SYSCLK)



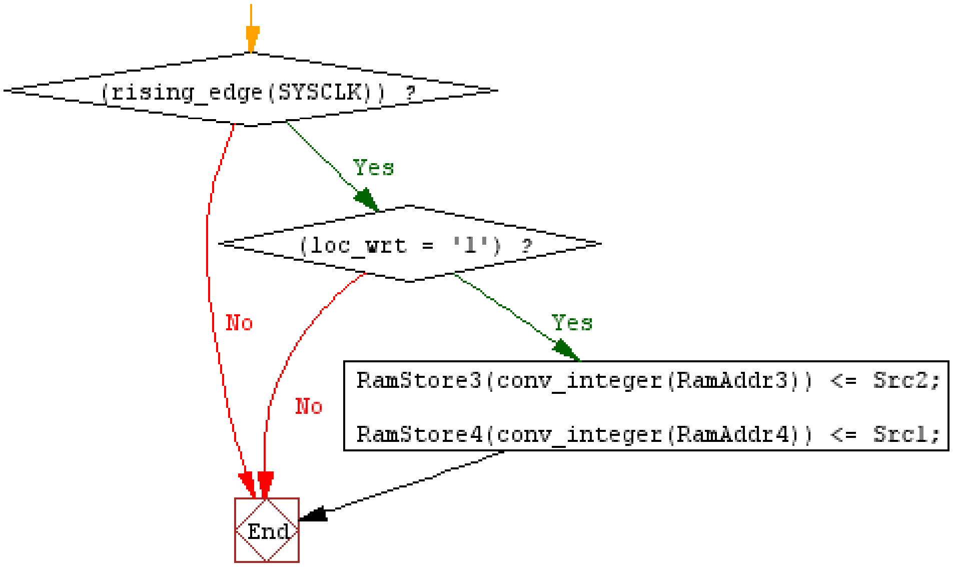
DataRamWrite1 : process (SYSCLK)



```
RamStore1(conv_integer(RamAddr1)) <= Src1;

RamStore2(conv_integer(RamAddr2)) <= Src2;
```

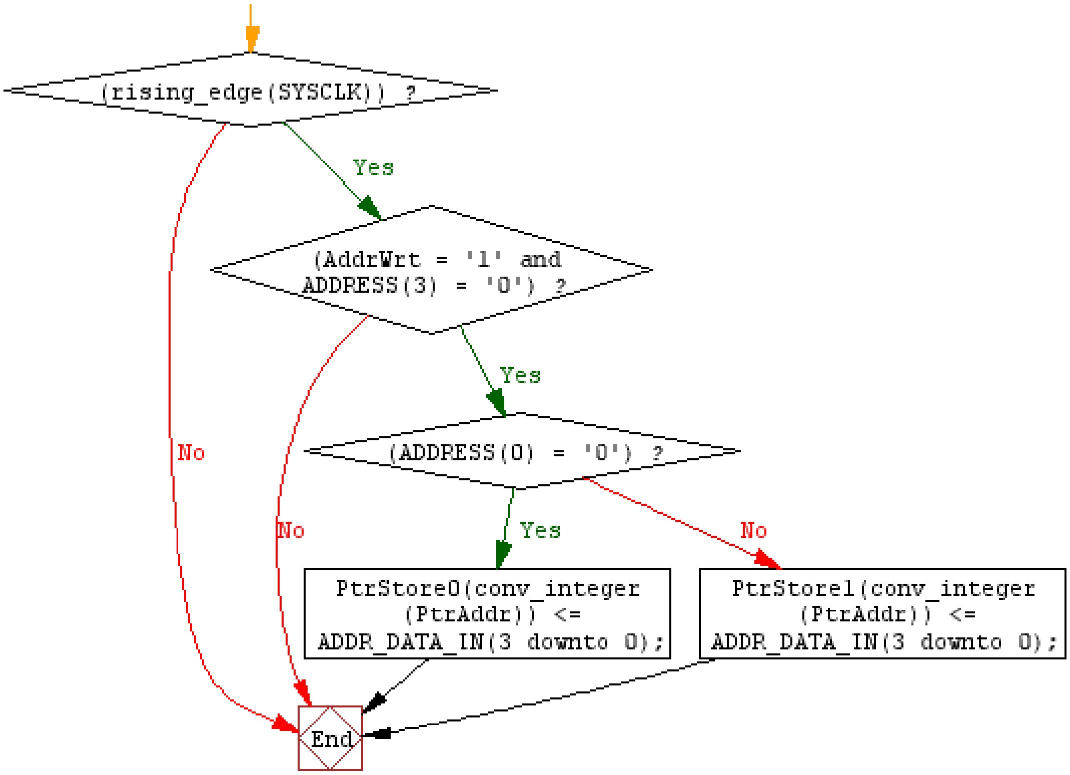
DataRamWrite2 : process (SYSCLK)



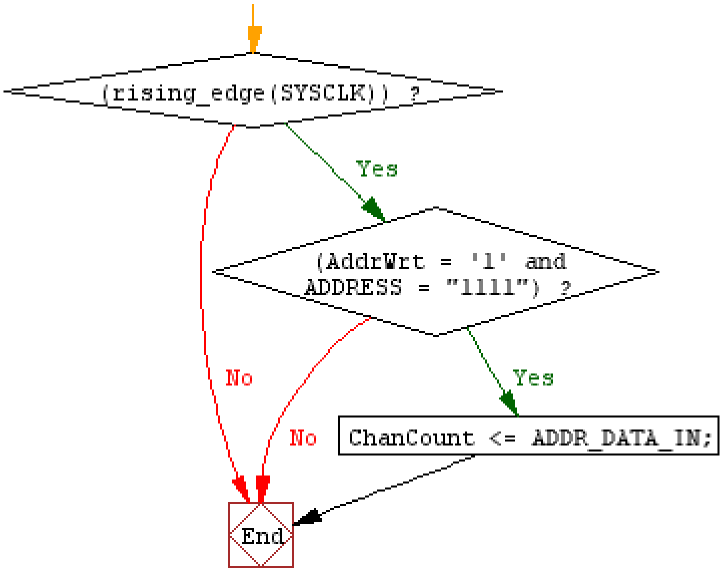
```
RamStore3(conv_integer(RamAddr3)) <= Src2;

RamStore4(conv_integer(RamAddr4)) <= Src1;
```

PtrRamWrite : process (SYSCLK)



DataCntReg : process(SYSCLK)


(rising_edge(SYSCLK)) ?
Yes
No
(AddrWrt = '1' and
ADDRESS = "1111") ?
Yes
No
ChanCount <= ADDR_DATA_IN;
End